

UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE TECNOLOGIA – CENTRO POLITÉCNICO
DEPARTAMENTO DE ENGENHARIA MECÂNICA – DEMEC
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA INDUSTRIAL 4.0

Hérík Dominschek
Rainerio Santana Lima
Igor Budnik

DRONES PARA PULVERIZAÇÃO AGRÍCOLA

CURITIBA
2020

Hérík Dominschek
Rainerio Santana Lima
Igor Budnik

DRONES PARA PULVERIZAÇÃO AGRÍCOLA

Monografia apresentada ao curso de especialização em Engenharia Industrial 4.0 do Departamento de Engenharia Mecânica, do Setor de Tecnologia, da Universidade Federal do Paraná, como requisito à obtenção do título de Especializa em Engenharia Industrial 4.0.

CURITIBA

2020

RESUMO

Drones tem sido um tópico muito popular sobre os conceitos 4.0 atual, porque essa tecnologia possui diferentes desempenhos para tarefas autônomas. Esse artigo foca na metodologia adotada para dar aos Drones a capacidade de se locomover de forma dinâmica, baseada em dados experimentais e possibilita novos métodos para sistemas dinâmicos e adaptativos. E será demonstrado onde essa tecnologia se enquadra em todo o desenvolvimento dos drones, para possibilitar voos coletivos. Também será descrito como esse sistema trabalha, como modelos lineares podem ser representados, controle inteligente de estabilidade e suas respectivas aplicações na eletrônica.

Neste contexto, este trabalho apresenta uma técnica para cálculo de trajetória através da utilização de um algoritmo *PSO (Particle Swarm Optimization)*. O algoritmo é um método de otimização inspirado na natureza e possui uma técnica baseada em população para problemas multidimensionais e multimodais, como o potencial de implementação de restrições no domínio de pesquisa.

Para controle dos drones, antes mesmo de desenvolver os controladores, é fundamental encontrar um modelo que representa o sistema, através de estimadores discretos. E o desempenho dos controladores PID (Proporcional – Integral - Derivativo) multivariáveis deverão mostrar uma resposta em malha fechada, usando os parâmetros de ganhos encontrados e aplicados, por uma abordagem de Redes Neurais e aerodinâmica, sob uma modelagem matemática de motores.

Palavras-chaves: Inteligência Coletiva. Identificação de Sistemas. Controle Inteligente.

LISTA DE ILUSTRAÇÕES

Figura: Procedimento de projeto do controlador	10
Figura 1: Malha de controle em malha fechada utilizada para controlar os motores.	13
Figura 2 :Sinal de controle medido aplicado ao circuito.	14
Figura 3: Velocidade medida após duas referências e perturbação na saída.....	14
Figura 4: Desempenho do modelo após a aproximação por mínimos quadrados. ..	15
Figura 5: Sinal de controle adaptativo direto encontrado.	16
Figura 6: Ganhos do PID Adaptativo (muda automático caso as entradas mudarem).	17
Figura 7: Constantes aproximadas por mínimos quadrados (em tempo real).	17
Figura 8: Algoritmo de inteligência coletiva. Calculo do percurso com base no melhor erro entre a distancia do obstáculo e a posição enviada aos Drones.	18
Figura 9: Experimento para identificação e comunicação entre os Drones.	20
Figura 10: Modelagem mecânica em Solidworks.....	20

SUMÁRIO

1 INTRODUÇÃO	6
2 REVISÃO BIBLIOGRÁFICA	7
2.1 Inteligência Coletiva	7
2.1.1 Características.....	7
2.1.2 Precursores	7
2.1.3 Vocabulário	8
2.1.4 Equações	9
2.2 Fundamentos e conceitos do controlador PID.....	9
2.2.1 Controle PID adaptativo direto	10
2.3 Apresentação à identificação de sistemas.....	11
2.3.1 <i>Data Analytics</i>	11
2.4 Modelagem do motor	12
3 RESULTADOS	13
REFERÊNCIAS.....	19
APÊNDICE	20

1 INTRODUÇÃO

A tecnologia vem envolvendo durante anos e tem dado para nós a possibilidade para construir e criar máquinas e dispositivos complexos. Nós estamos vivendo uma nova era de robôs eletrônicos, onde o uso de drones, em nossas rotinas diárias, tem aumentado. Além dessa tecnologia possibilitar explorar várias tarefas autônomas, podemos também aplicar em diversos ambientes, como no campo. Antes, muito conhecimento nas áreas de eletrônica, controle e mecânica, foi importante para entender, lá atrás, como esses sistemas robóticos funcionam e hoje podemos pensar mais além, em eliminar todos nossos esforços em pilotar um drone; e nesse artigo, mostraremos como faremos vários drones voarem de forma que cooperem entre si e a trabalhar sincronizados, com um único objetivo, pulverizar uma área no campo.

A sintonia dos quatros PID (dois em cada eixos) não é tarefa fácil, pois podem ocorrer interações entre as malhas de controle, ocasionando no comportamento correto. Portanto, a sistematização de metodologias para sintonia dos controladores em conjunto, é importante para sistemas com estas características. Seguindo essa ideia, podemos descrever como se comporta um sistema controlado digitalmente. O sinal de saída do processo é convertido de forma digital através de um conversor analógico - digital (A/D), realizada nos instantes de amostragem. O Drone interpreta o sinal convertido através de uma sequência, por um algoritmo de controle. O conversor digital - analógico (D/A) transforma uma sequência de amostras em um sinal analógico que comanda o processo. Esta conversão do sinal controlado, mantém - se constante entre dois instantes de amostragem. Neste caso, como os algoritmos de controle descrevem o processo nos instantes de amostragem, este deve ser discretizado. Os sistemas discretos processam sequências amostrais e, portanto, podem ser representados por equações a diferenças. Como a equação relaciona a saída com as entradas, e as saídas em instantes anteriores, o sistema passa a ser descrito por um modelo de entradas e saídas.

O restante do projeto é organizado da seguinte forma: os fundamentos e conceitos do problema, a metodologia ao uso do algoritmo de otimização, modelagem aerodinâmica nos motores e identificação usando *Analytics* (sistemas *Neuro – Fuzzy*) e controladores PID adaptativos diretos.

2 REVISÃO BIBLIOGRÁFICA

2.1 Inteligência Coletiva

A inteligência coletiva é a denominação de uma subárea da inteligência artificial que abrange um conjunto de metodologias e técnicas inspiradas na inteligência coletiva com base em algumas espécies animais como formigas, abelhas. E possui como propriedade de sistemas de agentes não – inteligentes de capacidade individual limitada o comportamento coletivo inteligente.

- Sistema auto – organizável
- Um agente contribui para o grupo tornando – se mais forte
- Interações locais entre eles frequentemente originam um padrão global
- Elementos não inteligentes têm comportamentos coletivos inteligentes

2.1.1 Características

- Número acentuado de elementos
- Controle descentralizado
- Membros autônomos se comportam com regras simples baseadas em informação local, interagindo com vizinhos mais próximos
- Membros colaboram para achar comportamentos globais complexos

2.1.2 Precursores

Na simulação de Boids de Craig W. Reynolds (1987), um modelo consiste em três regras:

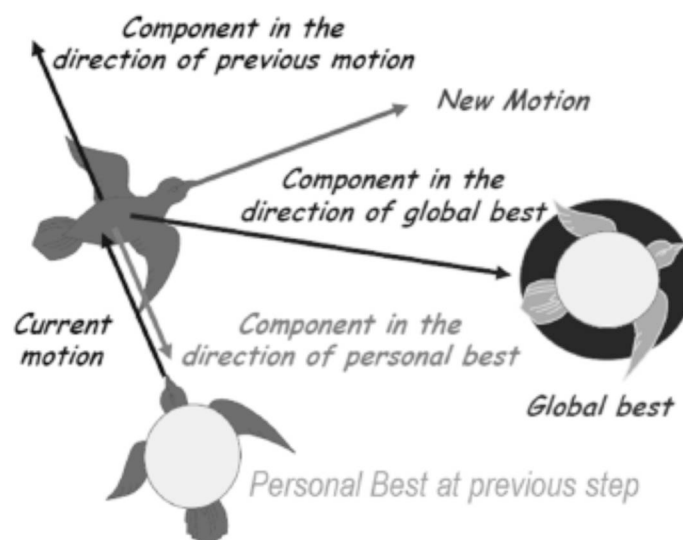
- Separação: evitar colisões de um boid com outro;
- Alinhamento de velocidade: tentar ir à mesma velocidade dos vizinhos no bando;
- Coesão: tentar mover para o centro do bando quando perceber onde está ele.

Para Heppner e Grenander (1990) existem regras que habilitam a um largo número de pássaros a juntarem – se de forma síncrona.

2.1.3 Vocabulário

Agente	Um único indivíduo no enxame
Posição	Coordenadas de um agente num espaço n-dimensional
Enxame	Toda a coleção de agentes
Aptidão	Um único número que representa a quão boa é uma solução dada (representado por uma posição)
$pbest$	A localização da melhor aptidão retornado para um determinado agente
$gbest$	A localização da melhor aptidão retornado para um enxame inteiro
V_{max}	A velocidade máxima permitida em uma direção

- Cada partícula conserva o conhecimento do seu melhor valor da função de aptidão denotada pela versão local, $pbest$ (*personal best*).
- Um outro melhor valor é seguido pela versão global, $gbest$ (*global best*) do PSO e sua localização obtida de alguma partícula que forma o enxame.



2.1.4 Equações

- **Equação 1:** Velocidade

$$V_{id} = \underbrace{w * v_{id}}_{\text{Inércia}} + \underbrace{C_1 * ud * (pbest_{id} - x_{id})}_{\text{Comportamento cognitivo}} + \underbrace{C_2 * Ud * (gbest_d - x_{id})}_{\text{Comportamento social}}$$

- **Equação 2:** Posição

$$x_{id} = x_{id} + \beta t * v_{id}$$

w : Fator de inércia

C_1 : Atração $pbest$

C_2 : Atração $gbest$

βt : Período de tempo

Ud : número com distribuição uniforme

$pbest_i$: Posição prévia com melhor aptidão

$gbest_i$: Número de partículas vizinhas

2.2 Fundamentos e conceitos do controlador PID

O algoritmo PID vem sendo usado com sucesso em vários processos atualmente. Esse tipo de controlador é uma arquitetura mais usada para controle automático. Foi a primeira solução experimentalmente quando a realimentação foi utilizada e atualmente abrange 90% das malhas em processos para fins acadêmicos e industriais (Ästrom e Hägglund, 2001). O controlador PID é utilizado em sistemas *SISO* (*Single Input, Single Output*), além do mais contém apenas uma variável controlada e uma manipulada. Então, muitos sistemas como esses são implementados simultaneamente em um processo e o desempenho pode ser afetado pela interação com outras malhas (Marlin, 1995). Nesse contexto, um sistema multivariável pode ser controlado por esses controladores em cada uma de suas malhas.

O controlador PID é composto por três parâmetros: Ação proporcional (K_p), a qual é proporcional à amplitude do valor do sinal de erro do processo. Sua atuação é capaz de melhorar a precisão do sistema reduzindo o erro de estado estacionário. Quando ajustado com valores grandes, o comportamento transitório tende a apresentar maiores oscilações. Logo, o aumento deste ganho é considerável apenas até determinados limites, para evitar à instabilidade do sistema (Ogata, 1998). A atuação da ação integral (K_i) diretamente ligada à melhoria da precisão do sistema e permite o seguimento de referência com o erro de estado estacionário nulo. Portanto, esta ação de controle não pode ser utilizada

separadamente, pois sua aplicação deve ser feita em conjunto com a ação proporcional (Ogata, 1998). Já na ação derivativa (K_d), a ação de controle atua apenas durante o período transitório, que proporciona correções na saída do sistema, e sua atuação não influencia o valor final de erro de estado estacionário (Ogata, 1998).

O modelo $G(s)$ de um controlador PID pode ser definido como (Ogata, 1998):

$$U(t) = K \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right]$$

Onde $e(t)$ corresponde ao sinal do erro.

2.2.1 Controle PID adaptativo direto

O projeto de controle adaptativo direto apresentado por Zu-zhi (1958) baseia – se no método de projeto por alocação de pólos (Coelho et al; 1996). Nesta técnica uma função de transferência desejada é definida pelo projetista e o procedimento para aplicar o método consiste em:

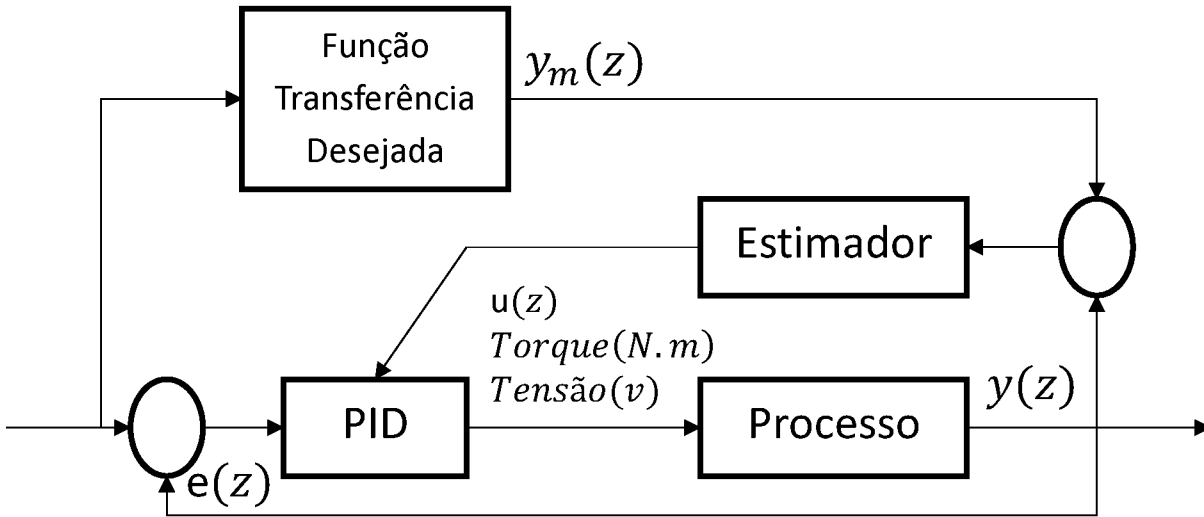
- I. Selecionar uma função de transferência desejada;
- II. Projetar o controlador tal que a função de transferência em malha fechada seja igual a função de transferência desejada;
- III. Estimar os parâmetros PID em tempo real através do método MQR.

$$R(q^{-1})\Delta u(k) = T(q^{-1})y_r(k + d) - S(q^{-1})y(k)$$

Conforme descrito por Zhu-zhi (1985), para o projeto do controlador a função de transferência em malha fechada deve ser igual a uma função de transferência desejada previamente especificada e dada por

$$\frac{y}{y_r} = \frac{y_m}{y_r} = \frac{S(Z^{-1})}{T(Z^{-1})}$$

Lembrando o controle RST que é um controlador baseado no método de alocação de pólos que permite a definição de diferentes especificações em relação à dinâmica desejada para os problemas de rastreamento e regulação.



2.3 Apresentação à identificação de sistemas

A identificação de sistemas é a determinação de um modelo matemático que represente o sistema, com base em entradas e saídas. Este modelo não passa de uma equação matemática que representa as características reais de um sistema, a qualquer instante de tempo. A identificação pode ser dividida em quatro etapas básicas: *i)* planejamento de perturbações, coleta e tratamento de dados; *ii)* seleção de estrutura e tipo de modelo; *iii)* identificação do modelo e *iv)* validação dos modelos encontrados (Machado, 2004).

No entanto, foi adotado o procedimento de identificação off – line para este projeto. O qual armazenam – se as medidas de entradas e saídas para aplicação e avaliação do algoritmo MQR (mínimos quadrados não - recursivos).

2.3.1 Data Analytics

Considere um processo físico caracterizado por uma entrada, $u(t)$, uma saída, $y(t)$, uma perturbação $e(t)$ e com função de transferência discreta linear da forma

$$A(Z^{-1})y(t) = Z^{-d}B(Z^{-1})u(t) + e(t)$$

E cuja representação por uma equação a diferenças é

$$y(t) = -a_1y(t-1) - \dots - a_nay(t-n) + b_0u(t-d) + \dots + b_nbu(t-d-nb) + e(t)$$

Dessa forma, definimos um modelo matemático que representa o comportamento do sistema de controle do Drone; e através de uma abordagem de regressão linear múltipla, definimos o vetor de medidas e parâmetros, matriz de observação e a estimativa com base nos dados coletados. A melhor função é dada através de uma função objetivo, ou seja, quanto menor o valor do somatório do erro melhor o modelo e apresenta a seguinte característica:

$$J = \frac{\sum_{t=1}^N [y(t) - \tilde{y}(t)]^2}{N}$$

2.4 Modelagem do motor

Antes de tudo precisamos estudar como os motores funcionam e qual é a combinação de forças, ao torque, fundamentais para determinar as direções de voo dos Drones.

Ainda, adotando uma metodologia mais comum em motores de corrente contínua, vamos falar um pouco sobre espaço de estado de cada circuito.

O circuito de armadura é regido por

$$e_a = R_m i_a + L_m \dot{i}_a + e_m$$

Onde R_m é a resistência de armadura, L_m é a indutância, i_a é a corrente, e e_m é a tensão induzida e e_a é a tensão aplicada na entrada do motor.

A tensão induzida depende da velocidade e é dada por

$$e_m = K_e \omega$$

Onde K_e é a constante e ω é a velocidade angular do eixo do motor.

O torque, produzido pelo motor, depende somente da corrente e através da carga mecânica, associada ao motor, o momento de inércia e o coeficiente de atrito viscoso, conseguimos encontrar uma matriz e suas componentes do vetor de estado.

3 RESULTADOS

Em busca dos melhores parâmetros para controlar o sistema, é preciso a priori um modelo que represente melhor o comportamento conforme as direções que foi projetada. Entretanto, estas direções, ou melhor dizendo, esta trajetória que foi determinada, é formada simplesmente aplicando referencias para determinar as respostas, em função do comportamento de controle, ou seja, inicialmente foi realizado a identificação do sistema em malha fechada. Quando aplicado entradas nas duas coordenadas, ajustado o ganho proporcional, para melhor oscilação, foi obtido as saídas e saídas foram lidas e salvas pelo ARDUINO.

Obtido os valores de entrada e saída, foram estimadas as constantes, através de mínimos quadrados, que melhor representou a equação à diferença. Como representa a equação 1 da seção 3.

$$\begin{aligned} y(k) = & 0.3316y(k-1) + 0.2804y(k-2) + 0.2464y(k-3) \\ & - 0.0118u(k-1) - 0.0034u(k-2) + 0.1556u(k-3) \end{aligned} \quad (1.3)$$

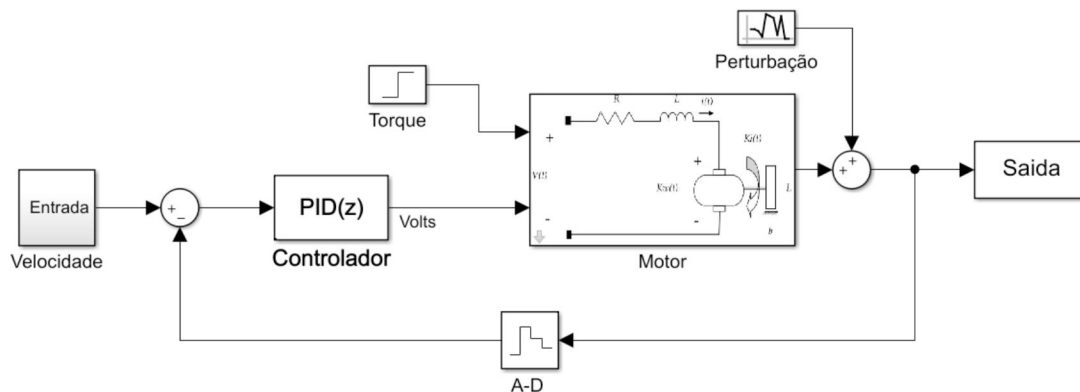


Figura 1: Malha de controle em malha fechada utilizada para controlar os motores.

Fonte: O autor. - Simulink/Matlab.

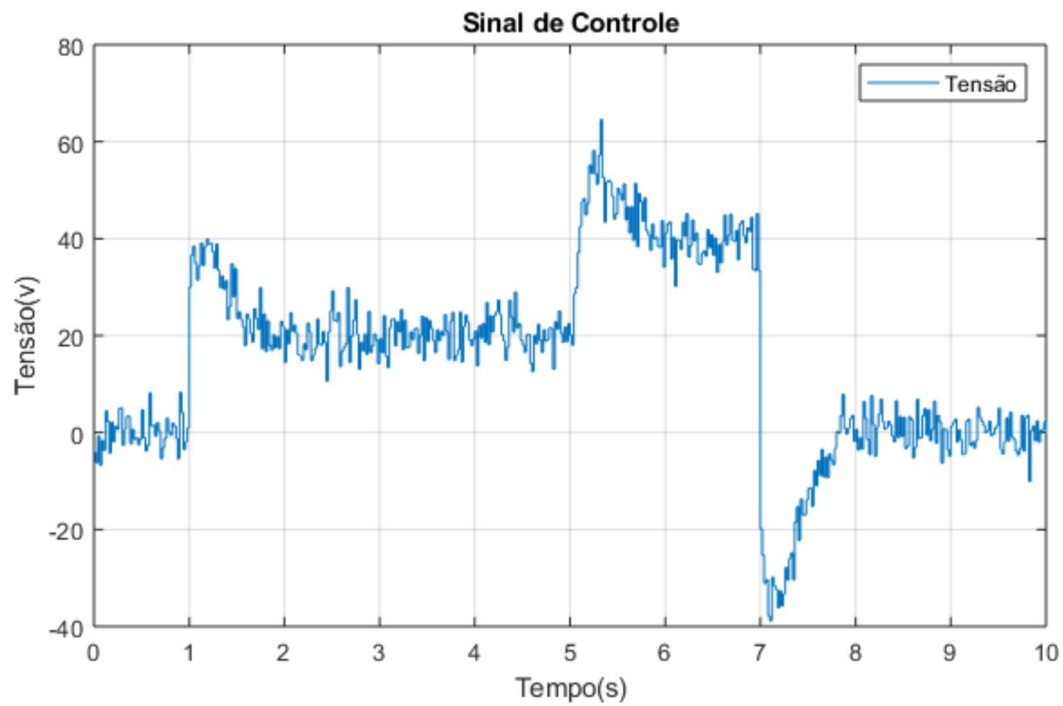


Figura 2 :Sinal de controle medido aplicado ao circuito.

Fonte: O autor.

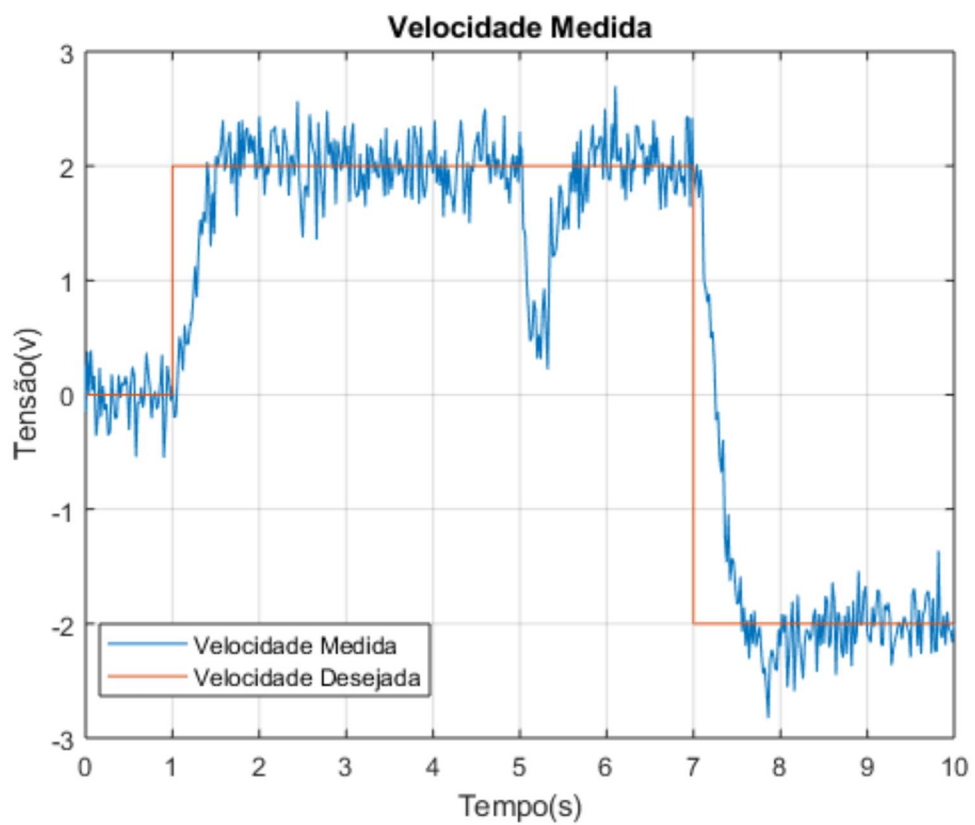


Figura 3: Velocidade medida após duas referências e perturbação na saída.

Fonte: O autor.

Observe, na figura 3, entre 5 e 6, no espaço de tempo, o gráfico mostra o comportamento após uma perturbação randômica. Estamos falando de condições climáticas, onde o vento, por exemplo, está sempre atuando no comportamento de controle.

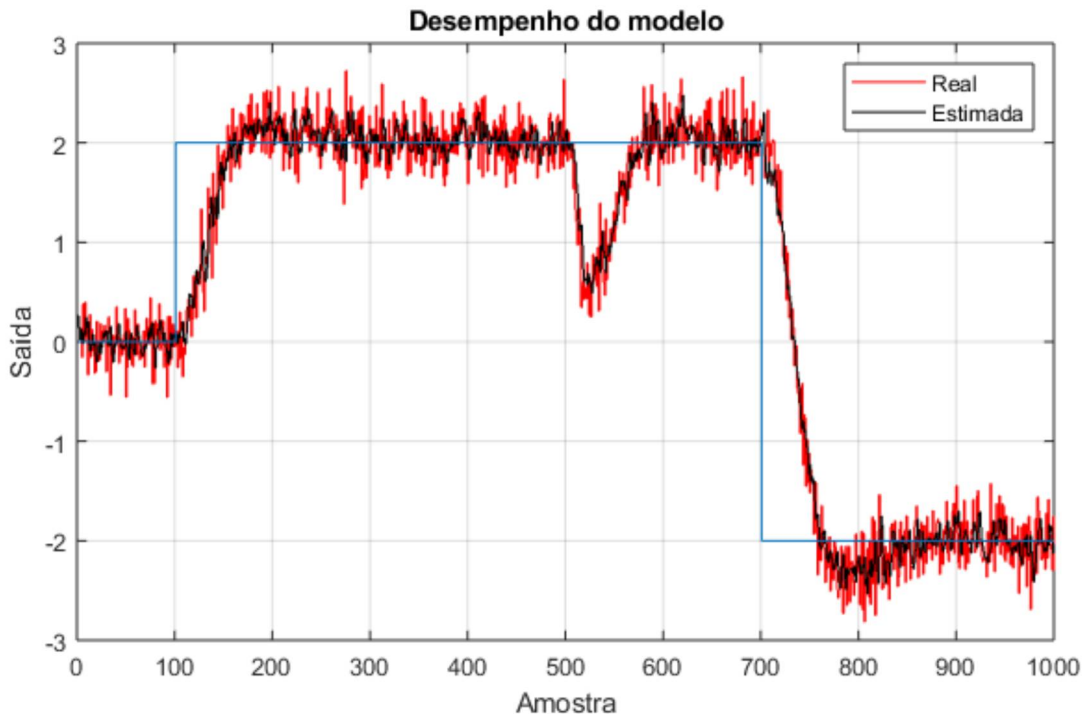


Figura 4: Desempenho do modelo após a aproximação por mínimos quadrados.

Fonte: O autor.

Observe, na figura 4, a curva estimada por mínimos quadrados; dessa maneira, podemos estimar as constantes de controle (parâmetros), já que agora temos a função de transferencia.

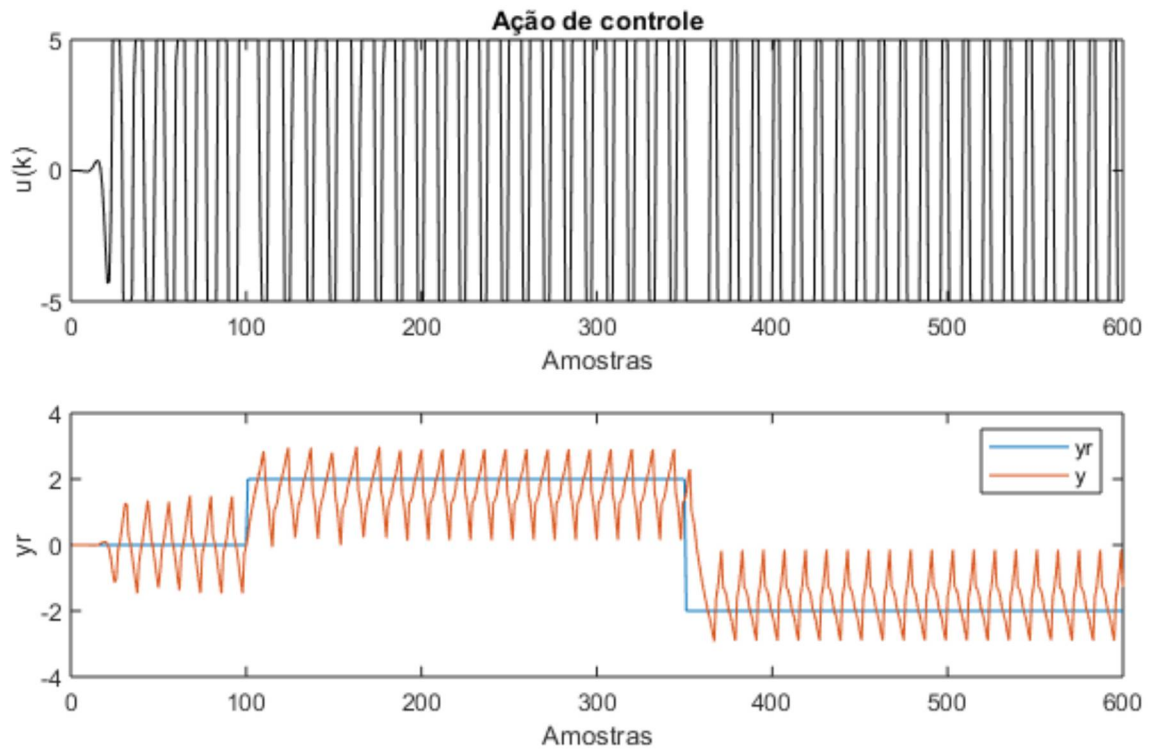


Figura 5: Sinal de controle adaptativo direto encontrado.

Fonte: O autor.

Neste instante, adaptamos o método de mínimos quadrados no controle adaptativo. Dessa maneira, o controle será responsável por estimar uma nova função de transferência e ganhos dos controlares em tempo real, com base em novas entradas e em função das condições climáticas. O controlador integral, é responsável por eliminar o erro em estado estacionário, ou seja, com esse controlador ajustado, o Drone sempre manterá estável, quando as perturbações tanto a próxima posição enviada pelo algoritmo de inteligência coletiva.

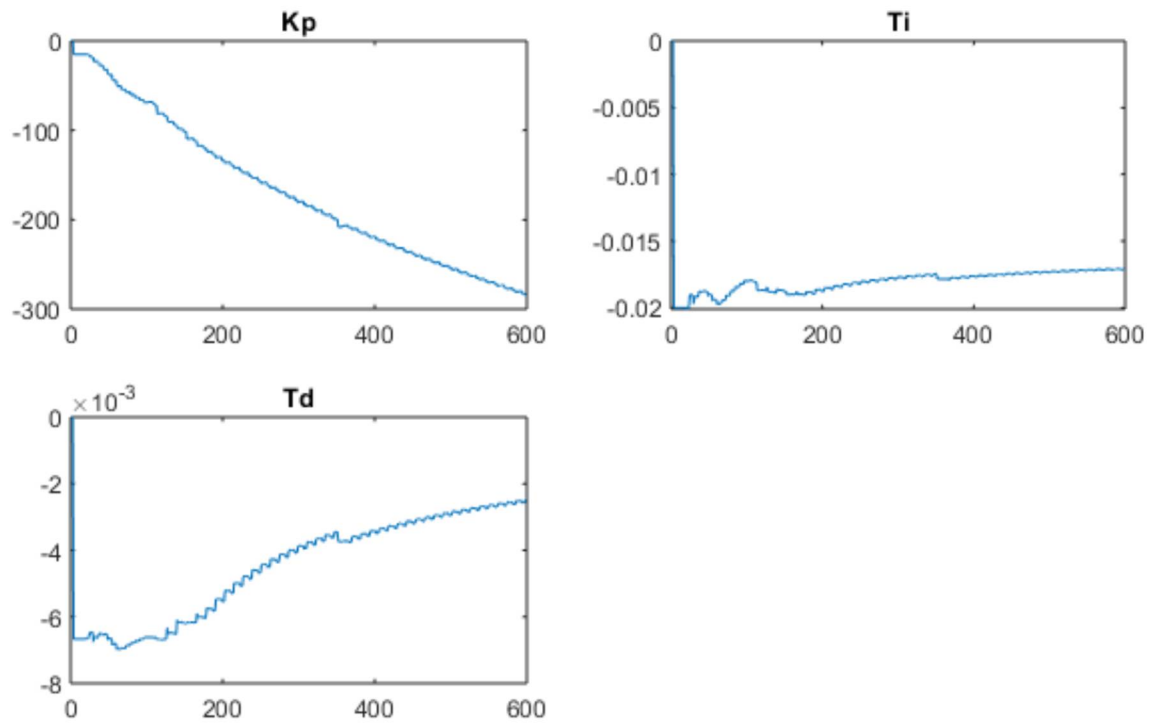


Figura 6: Ganhos do PID Adaptativo (muda automático caso as entradas mudarem).

Fonte: O autor.

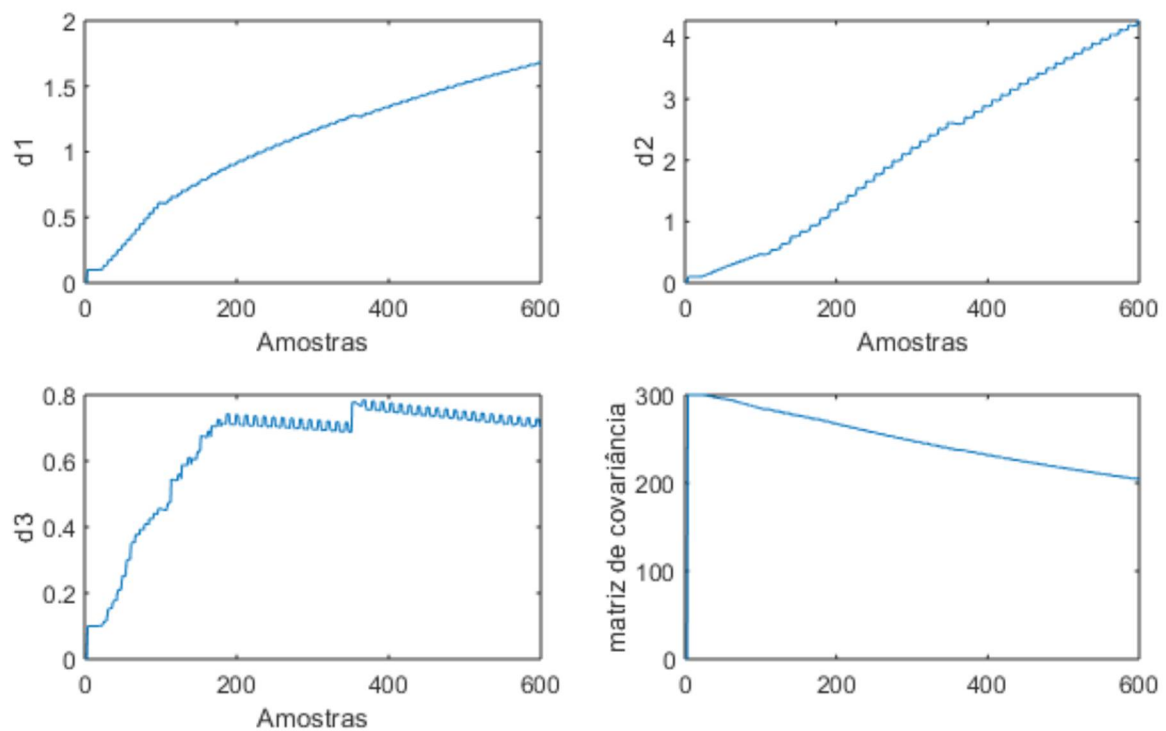


Figura 7: Constantes aproximadas por mínimos quadrados (em tempo real).

Fonte: O autor.

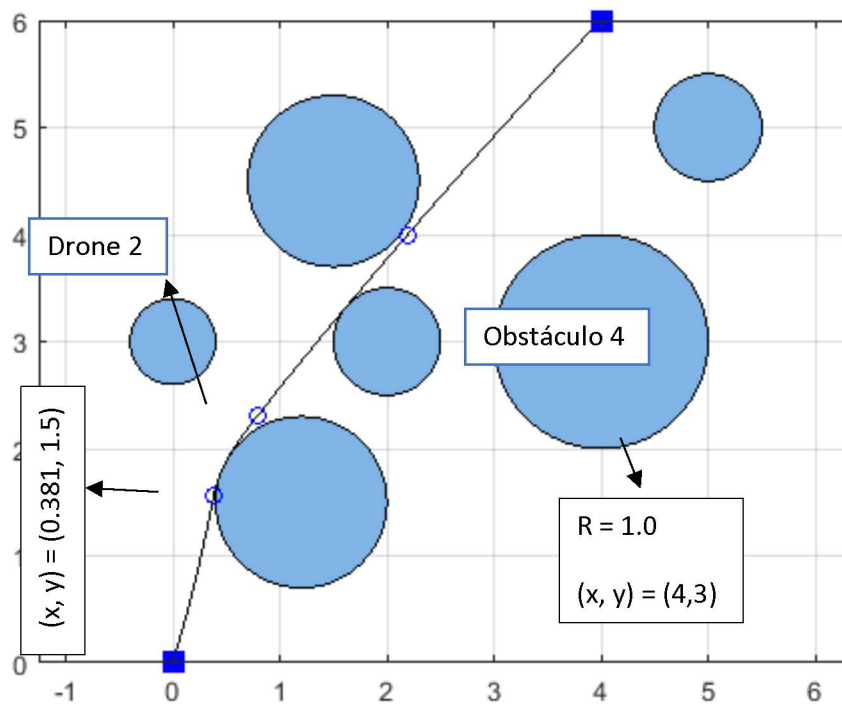


Figura 8: Algoritmo de inteligência coletiva. Cálculo do percurso com base no melhor erro entre a distância do obstáculo e a posição enviada aos Drones.

Fonte: O autor.

O planejamento de trajetórias foi adotado como um problema de otimização. Então uma sequência de posições que levem os drones da posição inicial à posição de destino será otimizada, também evitando colisões entre os drones e os obstáculos existentes, minimizando a distância a ser percorrida pelos drones.

Vamos considerar o problema em um espaço bidimensional, onde os drones são representados pelas suas coordenadas cartesianas, as posições inicial e final dos drones são conhecidas. E os obstáculos serão representados pelas coordenadas cartesianas de seu centro e pelo seu raio. Também os sensores serão utilizados para verificar a distância dos obstáculos em uma determinada direção e a cor dos obstáculos vai influenciar diretamente o raio de operação dos sensores.

Uma solução para explorar o ambiente previamente desconhecido. No entanto nos primeiros testes vamos realizar conhecendo o ambiente a priori. E então otimizar o percurso e os parâmetros dos controladores em malha fechada.

REFERÊNCIAS

Velasco, M. M. B. R. Sistemas neuro-fuzzy, ICA, PUC-Rio.

Gopal K. Dubey, — Fundamentals of Electrical Drives, Narosa Publishing House Pvt. Ltd., 2001, chap. 6.

J. Kennedy, — The Particle Swarm: Social Adaptation of Knowledge, Proceeding of the IEEE International Conference on Evolutionary Computation, ICEC1997, Indianapolis, pp. 303-308.

Benjamin C. Kuo, Farid Golnaraghi, 2009. Automatic Control Systems, 9th ed., John Wiley & Sons.

APÊNDICE

Figura 9: Experimento para identificação e comunicação entre os Drones.

Fonte: O autor.



Figura 10: Modelagem mecânica em Solidworks.

Fonte: O autor.